**TraceIt**

**Database Design Document**

**V 1.0**

**By**

|  |  |  |
| --- | --- | --- |
| 1 | Usman Ghani | Roll No 24 |
| 2 | Muhammad Shariq | Roll No 39 |
| 3 | Rayan Badar | Roll No 18 |



**Department of Computer Sciences**

**Namal University**

**Mianwali, Pakistan**

**Submission Date: 18th April, 2025**

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Approved by** |
| 22/04/24 | V 1.0 | This version includes updates based on the approved project proposal submitted earlier, covering complete ERD design, data dictionary, relationships, and especially in-text citations. | **Miss Asiya Batool** |

**TABLE OF CONTENTS**

[CHAPTER 1: PROJECT OVERVIEW 4](#_Toc195824026)

[1.1. INTRODUCTION: 4](#_Toc195824027)

[1.2. PROBLEM STATEMENT: 4](#_Toc195824028)

[1.3. PROJECT OBJECTIVES: 4](#_Toc195824029)

[1.4. DOCUMENT OBJECTIVES: 4](#_Toc195824030)

[CHAPTER 2: DETAILED DATABASE DESIGN 5](#_Toc195824031)

[2.1. ENTITY: 5](#_Toc195824033)

[2.2. DATA DICTIONARY: 5](#_Toc195824034)

[2.3. RELATIONSHIPS: 9](#_Toc195824035)

[2.4. ENTITY RELATIONSHIP DIAGRAM: 10](#_Toc195824036)

CHAPTER 3: [REFERENCES 11](#_Toc195824037)

# PROJECT OVERVIEW

## INTRODUCTION:

The **TraceIt** Lost & Found Management System is a mobile application backed by a cloud database, designed to digitize the lost-and-found process within institutions like universities or offices. It enables users to report lost or found items, automates match suggestions using algorithm-based logic, and allows administrators to verify and approve item claims. The system promotes transparency, improves recovery rates, and streamlines the item tracking process by eliminating traditional, inefficient, paper-based methods [1] [2].

## PROBLEM STATEMENT:

Lost and found items are frequently mismanaged [1] due to traditional manual logging systems that are inefficient, error-prone, and non-centralized. These systems lead to unclaimed items, delayed recoveries, and miscommunication. The lack of real-time notifications, proper verification, and data-driven tracking impairs the overall efficiency. The desired future state is a cloud-based mobile system that allows users to report, search, and recover items, while enabling administrators to verify and manage the overall process securely and efficiently [2].

## PROJECT OBJECTIVES:

The primary objective of TraceIt is to digitize and optimize the lost-and-found process by leveraging modern database management, automation, and secure authentication methods.

The key objectives include:

1. **Providing a centralized cloud-based database** for storing and managing lost and found item records.
2. **Enabling users to report lost items** and search for found objects through a user-friendly mobile application.
3. **Providing an automated matching mechanism** that suggests potential item matches based on entered details.
4. **Providing admins with verification tools** to validate reports and ensure authenticity.
5. **Integrating real-time notifications** to alert users about potential item matches.
6. **Generating analytical reports** to track lost-and-found trends and optimize recovery rates.

## DOCUMENT OBJECTIVES:

This document aims to:

1. **Define** the core database entities and their relationships.
2. **Provide** a detailed data dictionary for each table.
3. **Describe** the logical structure of the ERD based on system functionalities.
4. **Act as a** guide for database implementation and future modifications.
5. **Ensure** clarity, consistency, and maintainability of the data model.

# DETAILED DATABASE DESIGN



## ENTITY:

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Entity Name** | **Description** |
| 01 | USER | Represents individuals who use the app to report or recover items. |
| 02 | ADMIN | Authorized personnel responsible for verifying item matches and reports. |
| 03 | ITEM | Represents a physical object reported as lost or found. |
| 04 | REPORT | Contains lost or found reports submitted by users or admins. |
| 05 | VERIFICATION | Records the admin’s decision for verifying item claims. |
| 06 | MATCHINGLOG | Logs system-suggested item matches with a score and review status. |
| 07 | NOTIFICATION | Stores alerts sent to users/admins regarding match suggestions or actions. |
| 08 | FEEDBACK | Captures user feedback for item experience or system usage. |

## DATA DICTIONARY:

**1. USER:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | UserID | Integer | Primary Key | Unique identifier for user |
| 02 | Name | String | Not Null | Full name of the user |
| 03 | Email | String | Unique | Email address |
| 04 | PasswordHash | String | Not Null | Hashed password for security |
| 05 | ContactInfo | String | Optional | Phone number or other contact |

**2. ADMIN:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | AdminID | Integer | Primary Key | Unique identifier for the admin |
| 02 | Name | String | Not Null | Full name of the admin |
| 03 | Email | String | Unique, Not Null | Admin’s email for login/notifications |
| 04 | PasswordHash | String | Not Null | Encrypted password for authentication |
| 05 | ContactInfo | String | Optional | Contact number of the admin |

**3. ITEM:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| ItemID | Integer | Unique identifier |
| Category | String | Type/category of item |
| Description | String | Detailed explanation of the item |
| Color | String | Primary color of the item |
| Brand | String | Brand/manufacturer of the item |
| Size | String | Item size or dimensions |
| ItemCondition | String | Physical condition of the item |
| Material | String | Material composition (e.g., leather) |
| TagNumber | String | Unique label, barcode, or tag (optional) |
| Location | String | Where the item was reported |
| DateReported | DateTime | Timestamp of report |
| Status | String | Item state: 'pending', 'matched', etc. |
| ImageURL | String | Link to item image (optional) |

**4. REPORT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | ReportID | Integer | Primary Key | Unique ID for each report |
| 02 | ItemID | Integer | Foreign Key | References the reported item |
| 03 | ReporterID | Integer | Foreign Key | ID of user/admin who made the report |
| 04 | ReportType | String | Not Null | Type of report: ‘Lost’ or ‘Found’ |
| 05 | ReportedAt | DateTime | Not Null | Timestamp of when the report was filed |

**5. VERIFICATION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | VerificationID | Integer | Primary Key | Unique ID for each verification request |
| 02 | ItemID | Integer | Foreign Key | References the item under verification |
| 03 | AdminID | Integer | Foreign Key | Admin responsible for verifying the item |
| 04 | VerifiedAt | DateTime | Optional | Timestamp of verification action |
| 05 | Status | String | Not Null | ‘pending’, ‘approved’, or ‘rejected’ |

**6. MATCHING\_LOG:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | MatchID | Integer | Primary Key | Unique ID for each AI-based item match |
| 02 | LostItemID | Integer | Foreign Key | Refers to the item reported as lost |
| 03 | FoundItemID | Integer | Foreign Key | Refers to the item reported as found |
| 04 | MatchScore | Float | Not Null | AI-calculated score of similarity |
| 05 | MatchedAt | DateTime | Optional | Timestamp when the match was logged |
| 06 | Status | String | Not Null | Match status: ‘suggested’, ‘approved’, etc. |

**7. NOTIFICATION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | NotificationID | Integer | Primary Key | Unique ID for each notification |
| 02 | UserID | Integer | Foreign Key | ID of the user receiving the notification |
| 03 | AdminID | Integer | Foreign Key | ID of the admin receiving the notification |
| 04 | Message | String | Not Null | Content of the notification |
| 05 | Timestamp | DateTime | Not Null | When the notification was generated |
| 06 | SeenStatus | String | Not Null | Seen or Unseen status |

**8. FEEDBACK:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Name** | **Data Type** | **Constraint** | **Description** |
| 01 | FeedbackID | Integer | Primary Key | Unique identifier for each feedback entry |
| 02 | UserID | Integer | Foreign Key | User who gave the feedback |
| 03 | ItemID | Integer | Foreign Key (nullable) | Item related to feedback if applicable |
| 04 | FeedbackType | String | Not Null | Type: ‘item’, ‘system’, etc. |
| 05 | Comment | String | Optional | Text of the feedback |
| 06 | Rating | Integer | Range 1–5 | Numeric rating provided by the user |
| 07 | CreatedAt | DateTime | Not Null | Timestamp when feedback was submitted |

## RELATIONSHIPS:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Participating Entities** | **Relation** | **Business Rule** |
| 01 | User, Report | User creates Report | A user can report multiple items; each report belongs to one user. |
| 02 | Admin, Verification | Admin approves Verification | An admin verifies item claims through the verification module. |
| 03 | Item, MatchingLog | Item appears in Match | A single item can be part of multiple match logs (lost or found sides). |
| 04 | User, Notification | User receives Notification | A user can receive multiple notifications. |
| 05 | Admin, Notification | Admin receives Notification | An admin can receive multiple notifications. |
| 06 | User, Feedback | User gives Feedback | A user can give feedback on items or system experiences. |

## ENTITY RELATIONSHIP DIAGRAM:

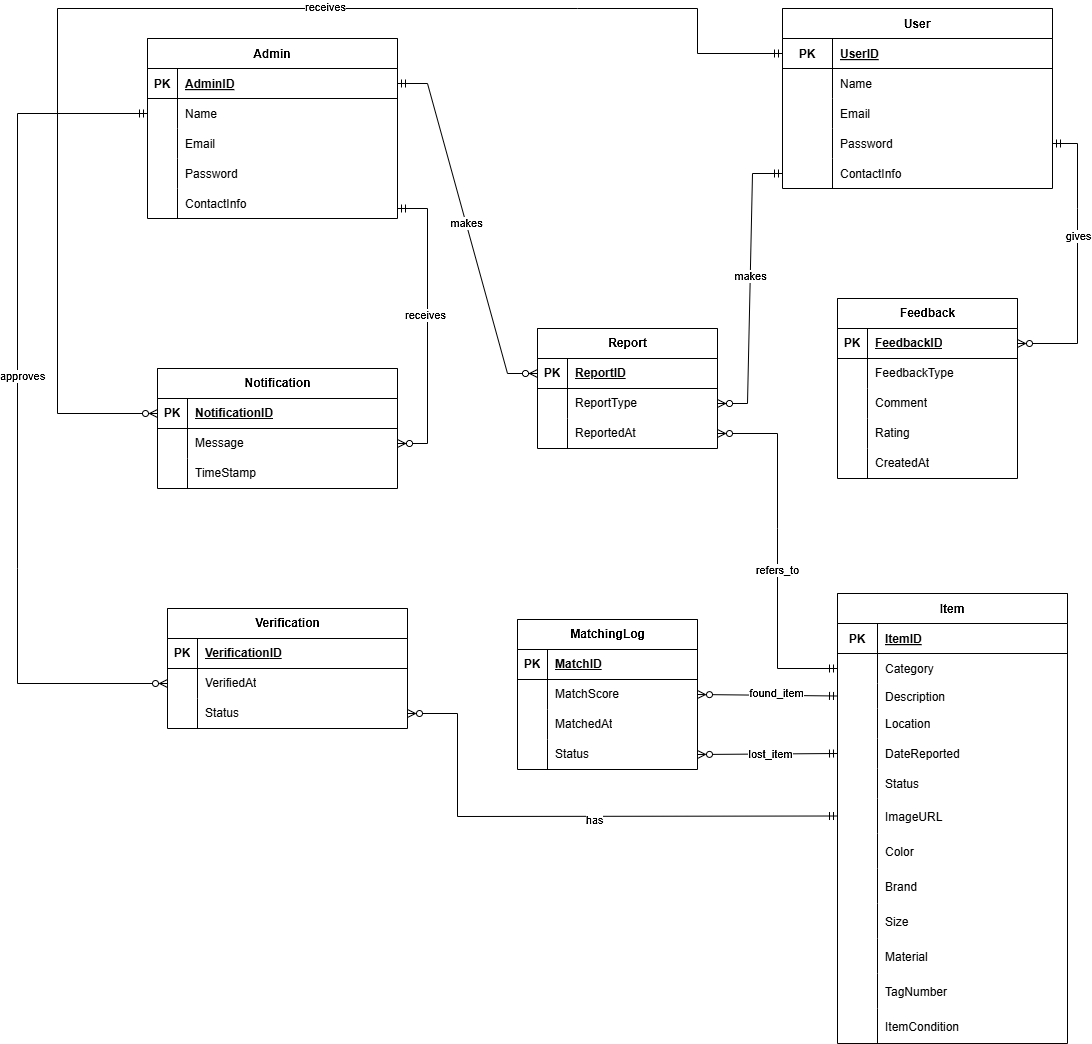


Figure 1: TraceIt – Entity Relationship Diagram (ERD)

### **2.4.1 Assumptions:**

The following assumptions were considered during the design of the Entity Relationship Diagram:

1. Admin and User are modeled as separate entities [1] due to their distinct roles and permissions.
2. An item can be verified multiple times to allow reappeals.
3. Feedback can optionally be linked to an item or be general (system-related).
4. A notification can be sent to either a user or an admin, depending on the event.

# CHAPTER 3: REFERENCES

|  |  |
| --- | --- |
| [1] | A. K. H. F. &. S. S. Silberschatz, Database System Concepts, McGraw-Hill, 2019. |
| [2] | Firebase, "Firebase Authentication Docs," 2024. [Online]. Available: https://firebase.google.com/docs/auth. [Accessed 2025]. |

**AI Tools Used:**

1. ChatGPT prompts were used to standardized project structure and refinement (used on dates 15-03-2025 to 17-04-2025).